WHAT IS CLAIMED IS:

1. A method for ensuring robustness of a natural language understanding (NLU) system comprising the steps of:

tagging recognized words of a command input to the NLU system to associate the command with a context;

translating the command to at least one formal command based on the tagged words;

determining a top ranked formal command based on scoring of the tagged recognized words and scoring translations of the at least one formal command;

determining whether the top ranked formal command is accepted by comparing a feature vector of the top ranked formal command to representations of feature vectors stored in an accept model; and

executing the top ranked formal command if accepted and preventing incorrect commands from execution to provide a robust NLU system.

2. The method as recited in claim 1, wherein the step of determining a top ranked formal command includes the step of ranking formal commands based on a product of scores of

the tagged words and scores of translations of the at least one formal command.

3. The method as recited in claim 1, wherein the step of determining a top ranked formal command includes the step of ranking N formal commands where N is a selectable system parameter.

4. The method as recited in claim 1, further comprising the step of determining whether the top ranked formal command is rejected by comparing the feature vector of the top ranked formal command to representations of feature vectors stored in a reject model.

5. The method as recited in claim 4, further comprising the step of providing the reject model by including representations of feature vectors of formal commands corresponding to words or sentences to be rejected.

6. The method as recited in claim 4, wherein the reject model includes a cluster of models.

7. The method as recited in claim 6, further comprising the step of clustering the cluster of models based on at least one of mistakes in commands, mistakes in arguments of the command, and processing mistakes.

8. The method as recited in claim 1, further comprising the step of providing the accept model by including representations of feature vectors of formal commands corresponding to words or sentences to be accepted.

9. The method as recited in claim 1, wherein the step of determining whether the top ranked formal command is accepted includes the step of computing a probability of acceptance for the command.

10. The method as recited in claim 9, wherein the step of computing a probability of acceptance for the command includes the steps of:

computing a probability of rejection for the command and comparing the probability of acceptance to the probability of rejection to determine if the command is to be executed.

11. The method as recited in claim 9, wherein the step of computing a probability of acceptance for the command includes the step of comparing the probability of acceptance to a threshold probability to determine if the command is to be executed.

12. The method as recited in claim 11, further comprising the step of modifying the threshold by the user.

13. The method as recited in claim 1, wherein the accept model includes a cluster of models.

14. The method as recited in claim 13, further comprising the step of clustering the cluster of models based on at least one of mistakes in commands, mistakes in arguments of the command, and processing mistakes.

15. The method as recited in claim 1, wherein the step of preventing incorrect commands, includes executing a do nothing command responsive to the incorrect commands.

16. A method for building an evaluation corpus for checking commands in a natural language understanding (NLU) system comprising the steps of:

providing a training corpus of words and sentences, the words and sentence having a user input form and a corresponding formal command associated with the user input form, at least some of corresponding formal commands including a do nothing command for one of incomplete and incorrect commands;

passing the words and sentences of the training corpus to the natural language understanding system to determine a top ranked command;

comparing the top ranked command to the corresponding formal command to determine if a match exists;

if a match exists, the word or sentence is placed in the accept corpus, otherwise in the reject corpus;

extracting features from the words or sentences of the accept corpus and the reject corpus to construct a feature vector for each word or sentence; and

constructing an accept model and a reject model, respectively, from the extracted feature vectors.

17.   The method as recited in claim 16, wherein the feature vectors include at least one of tagging scores for recognized words of the word or sentence represented by the feature vector and translation scores for formal commands associated with the word or sentence represented by the feature vector.

18.   The method as recited in claim 16, wherein the feature vectors include a do nothing score associated with words and sentences, the do nothing score for indicating a probability that the do nothing command is present for associated words and sentences.

19.   The method as recited in claim 16, wherein the feature vectors include a top command similarity measure for counting identical formal commands.

20.   The method as recited in claim 16, wherein the feature vectors include a parameter mismatch feature for measuring a number of command arguments in a translation of a command.

21. The method as recited in claim 16, further comprising the step of clustering feature vectors according to selected characteristics and conditions to provide at least one of a cluster of accept models and a cluster of

5   reject models.

22. The method as recited in claim 16, wherein the accept model and the reject model consist of mean vectors and covariance matrices of feature vectors representing the

10   words and sentences and a number of words and sentences stored in the model.

23. A program storage device readable by machine, tangibly embodying a program of instructions executable by

15   the machine to perform method steps for ensuring robustness of a natural language understanding (NLU) system, the method steps comprising:

tagging recognized words of a command input to the NLU system to associate the command with a context;

20   translating the command to at least one formal command based on the tagged words;

determining a top ranked formal command based on scoring of the tagged recognized words and scoring translations of the at least one formal command;

determining whether the top ranked formal command is accepted by comparing a feature vector of the top ranked formal command to representations of feature vectors stored in an accept model; and

executing the top ranked formal command if accepted and preventing incorrect commands from execution to provide a robust NLU system.

24. The program storage device as recited in claim 23, wherein the step of determining a top ranked formal command includes the step of ranking formal commands based on a product of scores of the tagged words and scores of translations of the at least one formal command.

25. The program storage device as recited in claim 23, wherein the step of determining a top ranked formal command includes the step of ranking N formal commands where N is a selectable system parameter.

26.   The program storage device as recited in claim 23, further comprising the step of determining whether the top ranked formal command is rejected by comparing the feature vector of the top ranked formal command to representations of feature vectors stored in a reject model.

27.   The program storage device as recited in claim 26, further comprising the step of providing the reject model by including representations of feature vectors of formal commands corresponding to words or sentences to be rejected.

28.   The program storage device as recited in claim 26, wherein the reject model includes a cluster of models.

29.   The program storage device as recited in claim 28, further comprising the step of clustering the cluster of models based on at least one of mistakes in commands, mistakes in arguments of the command, and processing mistakes.

30.   The program storage device as recited in claim 23, further comprising the step of providing the accept model by

including representations of feature vectors of formal

commands corresponding to words or sentences to be accepted.

31. The program storage device as recited in claim 23,

wherein the step of determining whether the top ranked

formal command is accepted includes the step of computing a

probability of acceptance for the command.

32. The program storage device as recited in claim 31,

wherein the step of computing a probability of acceptance

for the command includes the steps of:

computing a probability of rejection for the command

and comparing the probability of acceptance to the

probability of rejection to determine if the command is to

be executed.

33. The program storage device as recited in claim 31,

wherein the step of computing a probability of acceptance

for the command includes the step of comparing the

probability of acceptance to a threshold probability to

determine if the command is to be executed.

34.   The program storage device as recited in claim 33,

further comprising the step of modifying the threshold by

the user.

5   35.   The program storage device as recited in claim 23,

wherein the accept model includes a cluster of models.

36.   The program storage device as recited in claim 35,

further comprising the step of clustering the cluster of

10   models based on at least one of mistakes in commands,

mistakes in arguments of the command, and processing

mistakes.

37.   The program storage device as recited in claim 23,

15   wherein the step of preventing incorrect commands, includes

executing a do nothing command responsive to the incorrect

commands.

38.   A natural language understanding (NLU) system

20   comprising:

means for tagging recognized words of a command input

to the NLU system to associate the command with a context;

means for translating the command to at least one formal command based on the tagged words;

a robustness checker for determining a top ranked formal command based on scoring of the tagged recognized words and scoring translations the at least one formal command, the robustness checker for determining whether the top ranked formal command is accepted by comparing a feature vector of the top ranked formal command to representations feature vectors stored in an accept model; and

a command executor for executing the top ranked formal command if accepted and preventing incorrect commands from execution to provide a robust NLU system.

39. The system as recited in claim 38, wherein the top ranked formal command is ranked based on a product of scores of the tagged words and scores of translations of the at least one formal command.

40. The system as recited in claim 38, wherein the top ranked formal command is selected from a ranked list of N formal commands where N is a selectable system parameter.

41. The system as recited in claim 38, further comprising an accept model for storing representations feature vectors of accept commands, the representations of feature vectors being employed to compare with a user input command to determine an acceptance probability of formal commands.

42. The method as recited in claim 41, wherein the accept model includes a cluster of models.

43. The system as recited in claim 38, further comprising a reject model for storing representations of feature vectors of reject commands, the feature vectors being employed to compare with a user input command to determine a rejection probability of formal commands.

44. The system as recited in claim 43, wherein the reject model includes a cluster of models.

45. The system as recited in claim 38, wherein the robustness checker includes a feature extractor for extracting feature vectors from the command.

Y0999-331 (8728-310)   -44-

46. The system as recited in claim 38, wherein the feature vectors include tagging scores for recognized words of a word or sentence represented by the feature vector and translation scores for formal commands associated with the word or sentence represented by the feature vector.

47. The system as recited in claim 38, wherein the feature vectors include a do nothing score associated with words and sentences, the do nothing score for indicating a probability that a do nothing command is present for associated words and sentences, the do nothing command being associated with incomplete or incorrect commands which may be input as a command.

48. The system as recited in claim 38, wherein the feature vectors include at least one of a top command similarity measure for counting identical formal commands and a parameter mismatch feature for measuring a number of command arguments in a translation of a command.

49. The system as recited in claim 38, wherein the robustness checker includes a robustness evaluator for

Y0999-331 (8728-310)   -45-

determining whether the top ranked formal command is
accepted by comparing the top ranked formal command to
feature vectors stored in an accept model.

5   50. A natural language understanding system
comprising:

a corpus of rejectable commands corresponding to
incorrect commands capable of being input by a user; and

do nothing commands corresponding to the incorrect
10 commands input to the system which at least one of maintain
the system in an idle state and prompt the user to input a
response.